

# Weekly Report

---

04/01/2017-08/01/2017

---

Han Dongming Jan. 08 2017

## Work

1. Realize the "Drag and Drop Collapsible Tree Layout"

## To do

1. Use the layout on the project
2. Prepare the English and professional courses examination

## The simplified version code of the "Drop Collapsible Tree Layout"

<https://github.com/woaiwodib107/drag>

```
d3.json('data.json',function(error,data){
  if (error) throw error;
  var treeStart=function(data) {
    var width=1500,height=1500
    var svg=d3.select('#tree-container')
      .append('svg')
      .attr('width',width)
      .attr('height',height)
      .append('g')
      .attr("transform","translate(" + 10 + "," + 0 + ")")
    var tree = d3.tree()
      .size([1300, 1300]);
    var root = d3.hierarchy(data);
    var diagonal = function(d) {
      return "M" + d.source.y + "," + d.source.x
        + "C" + (d.source.y + d.target.y) / 2 + "," + d.source.
x
        + " " + (d.source.y + d.target.y) / 2 + "," + d.target.
x
        + " " + d.target.y + "," + d.target.x;
    }

    // 隐藏
```

```

var id=0
var disFa=function(root){
  root.id=id++
  if(root.hasOwnProperty('children')){
    root._children=root.children
    root.children.forEach(function(d){
      disFa(d)
    })
    delete(root.children)
  }
}
disFa(root)

//展开第一层
root.children=root._children
var nodeClick=function(d){
  if(!d.hasOwnProperty('_children') || !d.depth){
    return //此为叶子节点 或者 为根节点
  }
  if(d.hasOwnProperty('children')){//展开变折叠
    disFa(d)
  }else{//折叠展开
    d.children=d._children
  }
  update()
}
f={}
linkf={}
var update=function(){
  var links=root.links()
  var nodes=root.descendants()
  tree(root);
  nodes[0].parent={x:nodes[0].x,y:nodes[0].y}
  var link = svg.selectAll(".link")
    .data(links)

  link.enter()
    .append("path")
    .attr("class", "link")

  link.exit()
    .remove();

  d3.selectAll('.link')
    .transition()
    .duration(500)
    .attr("d", function(d) {
      linkf[d.id]=diagonal(d)
      return linkf[d.id]
    })
  var node = svg.selectAll(".node")

```

```

        .data(nodes,function(d){
            return d.id
        })
        // .attr("class", function(d) { return "node" + (d.children ? " node--internal" : " node--leaf"); })

        node.enter()
            .append("g")
            .attr('class','node')
            .attr('id',function(d){
                return d.id
            })
            .attr("transform",function(d){
                if(!f.hasOwnProperty(d.id))
                    return "translate(" + (d.parent.y) + "," + (d.parent.x) + ")";
                return "translate(" + (f[d.id].y) + "," + (f[d.id].x) + ")";
            })
            .append("circle")
            .attr("r", 10)
            .on('click',function(d){
                return nodeClick(d)
            })
        node.exit()
            .remove();

        var newNode=d3.selectAll('.node')
            .transition()
            .duration(500)
            .attr("transform", function(d) {
                f[d.id]={x:d.x,y:d.y}
                return "translate(" + (d.y) + "," + (d.x) + ")";
            })

    }
    update()
}
treeStart(data)
})

```

